High Level Architecture Interface Specification Version 1.3

Integrated Training Program

Defense Modeling & Simulation Office
(703) 998-0660 Fax (703) 998-0667
hla@msis.dmso.mil
http://www.dmso.mil/

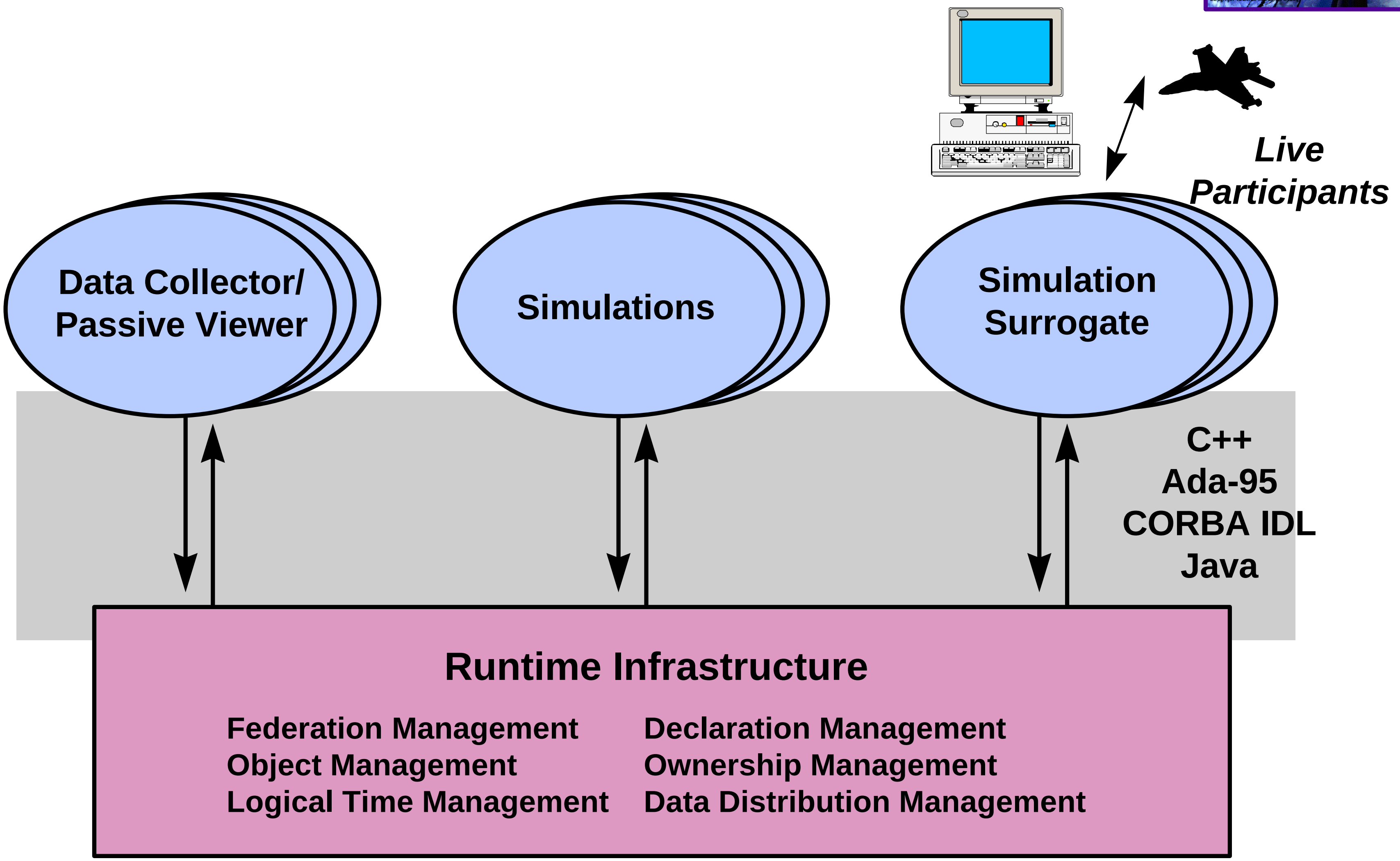
Current as of 1 April 98

High Level Architecture

- Major functional elements, interfaces, and design rules, pertaining to all DoD simulation applications, and providing a common framework within which specific system architectures can be defined

- HLA is the Technical Architecture for DoD Simulations

Functional View of the Architecture



Live Participants

Data Collector/ Passive Viewer

Simulations

Simulation Surrogate

C++
Ada-95
CORBA IDL
Java

Runtime Infrastructure

Federation Management
Object Management
Logical Time Management

Declaration Management
Ownership Management
Data Distribution Management

Rationale for an Interface Specification

- Provides a specification of the functional interfaces between federates and the RTI
- Facilitates, through a common, well defined, consistent set of interface definitions;
    - INTEROPERABILITY among simulations within a federation, and across functional M&S communities
    - REUSE of simulation components across federations, functional M&S communities, and RTIs

Interface Specification

- Provides a specification of the functional interfaces between federates and the RTI
  - Interfaces are divided into six service groups
- Each service specification includes:
  - Name and Descriptive Text
  - Supplied Arguments
  - Returned Arguments
  - Pre-conditions
  - Post-conditions
  - Exceptions
  - Related Services
- Application Programmer Interfaces (APIs) in CORBA IDL, C++, Ada ’95 and Java

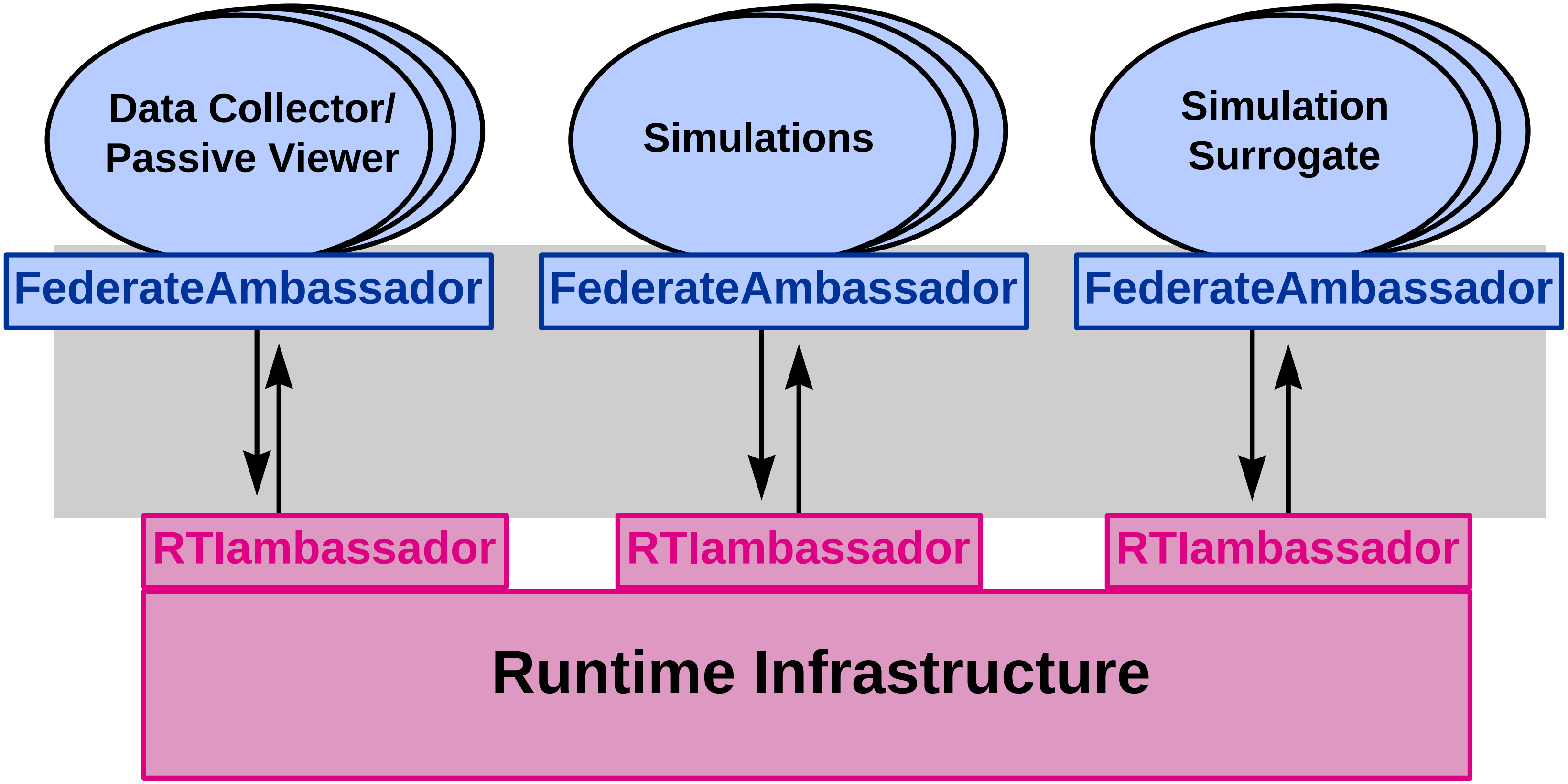
Six HLA Runtime Infrastructure Service Groups

Federation Management	(20 services)
Declaration Management	(12 services)
Object Management	(17 services)
Ownership Management	(16 services)
Time Management	(23 services)
Data Distribution Management	(13 services)

The Interface Specification also includes:

- Support Services (29 services)
- Management Object Model
- Federation Execution Data (FED)
- Application Programmers Interfaces (APIs)
- Harel state charts

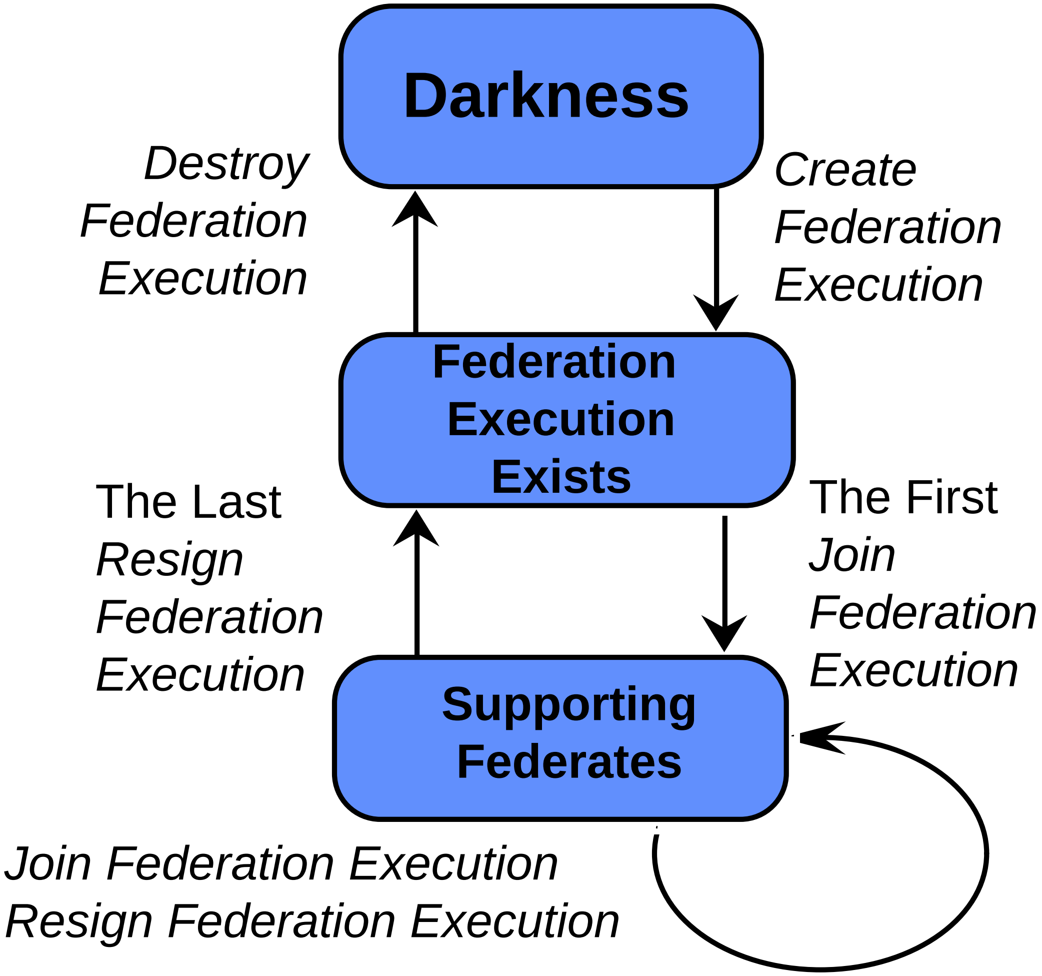
The RTIAmbassador and FederateAmbassador



Federation Management

- Coordinate federation-wide activities throughout the life of a federation execution
    - Used by federates to manage a federation execution to meet their needs
    - Includes Federation Execution Data (FED)
        - Defines class, attribute, interaction, parameter, and space names
        - Provides default transportation and ordering for attributes and interactions
- Interface functions include
    - Creation and destruction of a federation execution
    - Joining and resigning of a federate
    - Coordination of federation synchronization points
    - Coordination of federation Save and Restore

Federation Management


Darkness
Destroy Federation Execution
Create Federation Execution
Federation Execution Exists
The Last *Resign Federation Execution*
The First *Join Federation Execution*
Supporting Federates
Join Federation Execution
Resign Federation Execution

Federation Management Services

4.2 Create Federation Execution

4.3 Destroy Federation Execution

4.4 Join Federation Execution

4.5 Resign Federation Execution

4.6 Register Federation Synchronization Point

4.7 Confirm Synchronization Point Registration †

4.8 Announce Synchronization Point †

4.9 Synchronization Point Achieved

4.10 Federation Synchronized †

4.11 Request Federation Save

4.12 Initiate Federate Save †

4.13 Federate Save Begun

4.14 Federate Save Complete

4.15 Federation Saved †

4.16 Request Federation Restore

4.17 Confirm Federation Restoration Request †

4.18 Federation Restore Begun †

4.19 Initiate Federate Restore †

4.20 Federate Restore Complete

4.21 Federation Restored †

Declaration Management

- **Allow federates to specify the types of data they will send or receive by object class and attribute name and by interaction class from the FOM**
- **Interface functions include specification of:**
  - **Types of data to be sent (*PUBLISHING*):**
    - **Object classes and attributes and interaction classes that the federate is able to update or send**
  - **Types of data to be received (*SUBSCRIPTION*):**
    - **Object classes and attributes and interaction classes that the federate is interested to receive**
  - **Notifications to federates of relevance of object and interaction classes within the federation**
    - **Feedback to the federates from the RTI when registration of objects and sending of interactions should start or stop based on the subscriptions of other federates to those classes**

Declaration Management Services

5.2 Publish Object Class

5.3 Unpublish Object Class

5.4 Publish Interaction Class

5.5 Unpublish Interaction Class

5.6 Subscribe Object Class Attributes

5.7 Unsubscribe Object Class

5.8 Subscribe Interaction Class

5.9 Unsubscribe Interaction Class

5.10 Start Registration For Object Class †

5.11 Stop Registration For Object Class †

5.12 Turn Interactions On †

5.13 Turn Interactions Off †

Object Management

- Supports life cycle activities of objects and interactions
- Supports creation, modification, and deletion of objects, their attributes and the interactions they produce
- Interface functions include:
  - Registering and discovering objects
  - Updating and reflecting object attributes
  - Sending and receiving interactions
  - Deleting and removing objects
  - Changing default transportation preferences
  - Notification of in-scope and out-of-scope attributes
  - Requests for attribute value updates
  - Notifications to federates to start/stop updating attributes
    - Feedback to the federates from the RTI when attribute updates for a particular object instance should or should not be provided, given the relevance of those instance attributes to other federates

Object Management Services

6.2 Register Object Instance

6.3 Discover Object Instance †

6.4 Update Attribute Values

6.5 Reflect Attribute Values †

6.6 Send Interaction

6.7 Receive Interaction †

6.8 Delete Object Instance

6.9 Remove Object Instance †

6.10 Local Delete Object Instance

6.11 Change Attribute Transportation Type

6.12 Change Interaction Transportation Type

6.13 Attributes In Scope †

6.14 Attributes Out Of Scope †

6.15 Request Attribute Value Update

6.16 Provide Attribute Value Update †

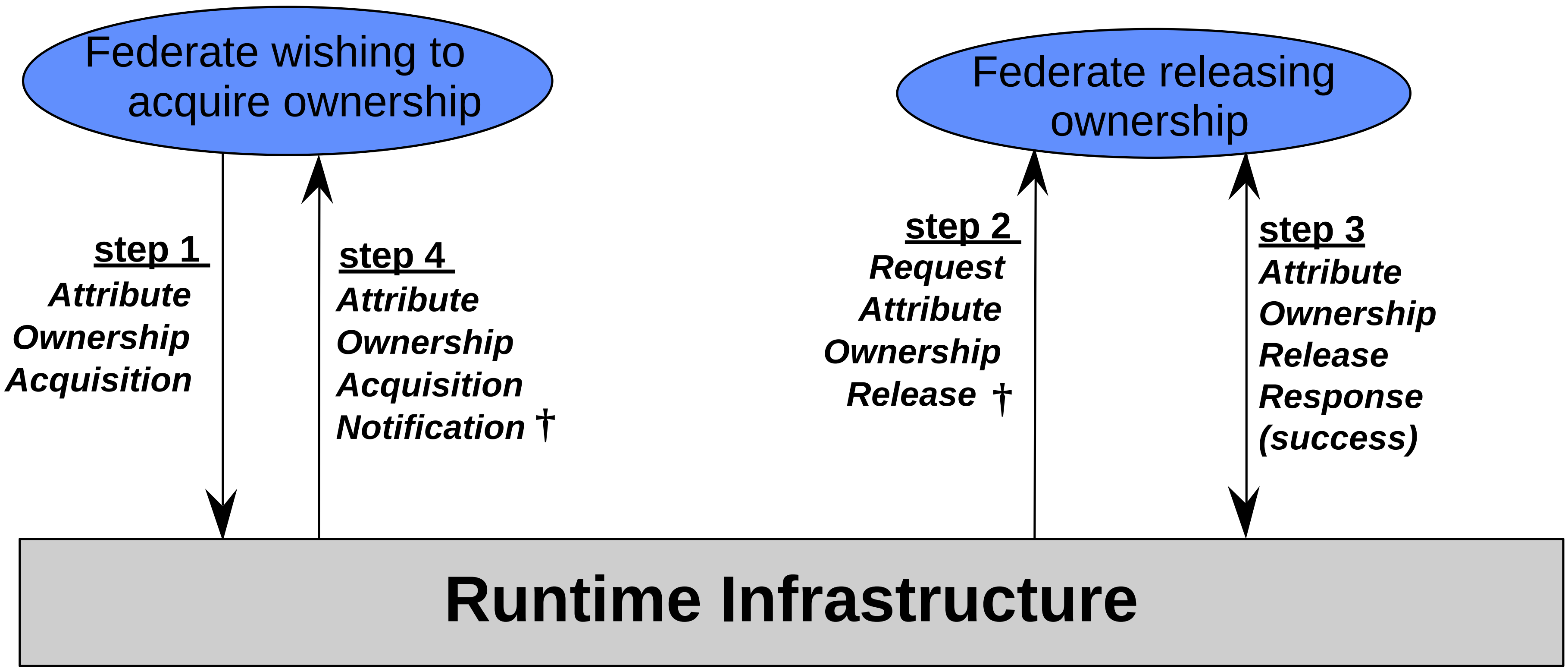
6.17 Turn Updates On For Object Instance †

6.18 Turn Updates Off For Object Instance †

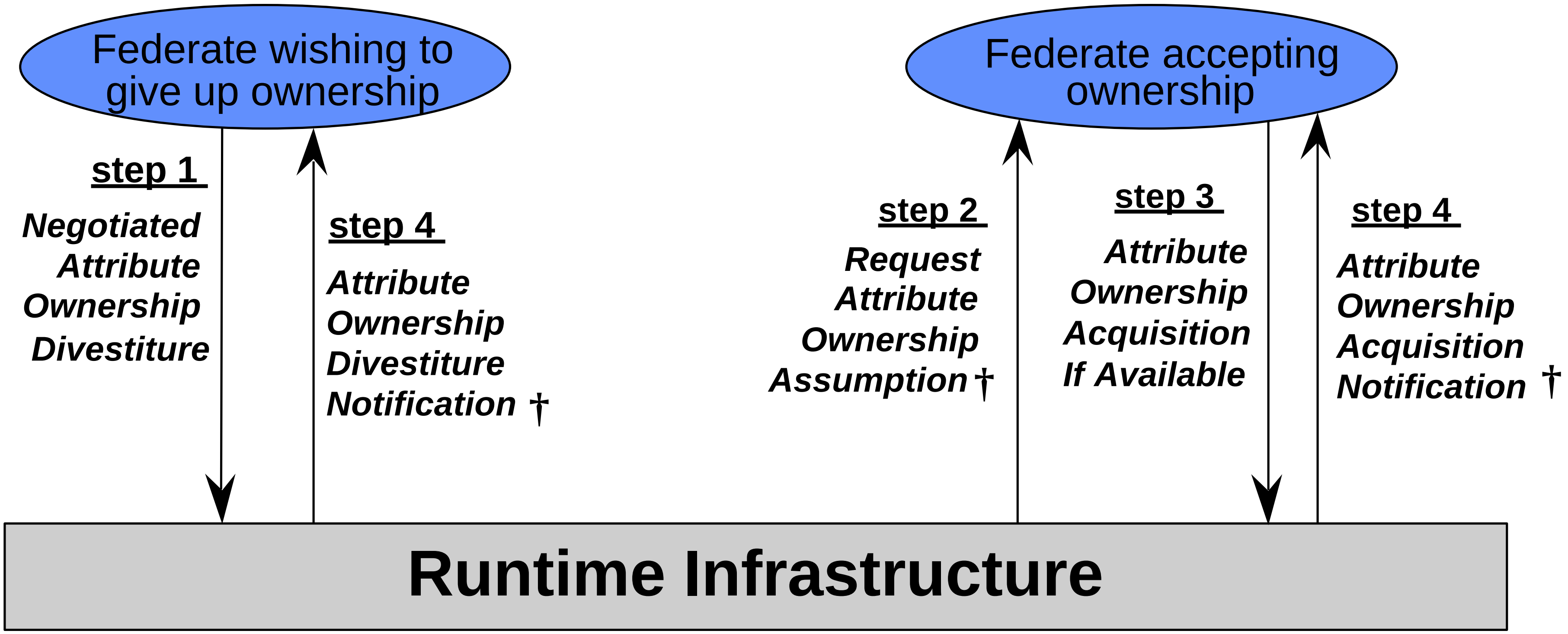
Ownership Management

- Allow federates to transfer ownership of object attributes
  - Federates transfer ownership based on federation execution design plans
  - RTI arbitrates transactions so that ownership is held by at most one federate at any time
  - Offers both ‘push’ or ‘pull’ based transactions
  - Acquisition requires current publication declarations for attribute
  - Ownership acquisition attempts can be both ‘invasive’ or based on ‘opportunity’
- Interface functions include
  - Attribute Ownership Divestiture (unconditional and negotiated)
  - Attribute Ownership Acquisition (explicit and if available)
  - Query Attribute Ownership

Ownership Acquisition


Federate wishing to acquire ownership
Federate releasing ownership
step 1
Attribute Ownership Acquisition
step 4
Attribute Ownership Acquisition Notification †
step 2
Request Attribute Ownership Release †
step 3
Attribute Ownership Release Response (success)
Runtime Infrastructure

Ownership Divestiture
(Negotiated)



Federate wishing to give up ownership
Federate accepting ownership
step 1
Negotiated Attribute Ownership Divestiture
step 4
Attribute Ownership Divestiture Notification †
step 2
Request Attribute Ownership Assumption †
step 3
Attribute Ownership Acquisition If Available
step 4
Attribute Ownership Acquisition Notification †
Runtime Infrastructure

Ownership Management Services

7.2 Unconditional Attribute Ownership Divestiture

7.3 Negotiated Attribute Ownership Divestiture

7.4 Request Attribute Ownership Assumption †

7.5 Attribute Ownership Divestiture Notification †

7.6 Attribute Ownership Acquisition Notification †

7.7 Attribute Ownership Acquisition

7.8 Attribute Ownership Acquisition If Available

7.9 Attribute Ownership Unavailable †

7.10 Request Attribute Ownership Release †

7.11 Attribute Ownership Release Response

7.12 Cancel Negotiated Attribute Ownership Divestiture

7.13 Cancel Attribute Ownership Acquisition

7.14 Confirm Attribute Ownership Acquisition Cancellation †

7.15 Query Attribute Ownership

7.16 Inform Attribute Ownership †

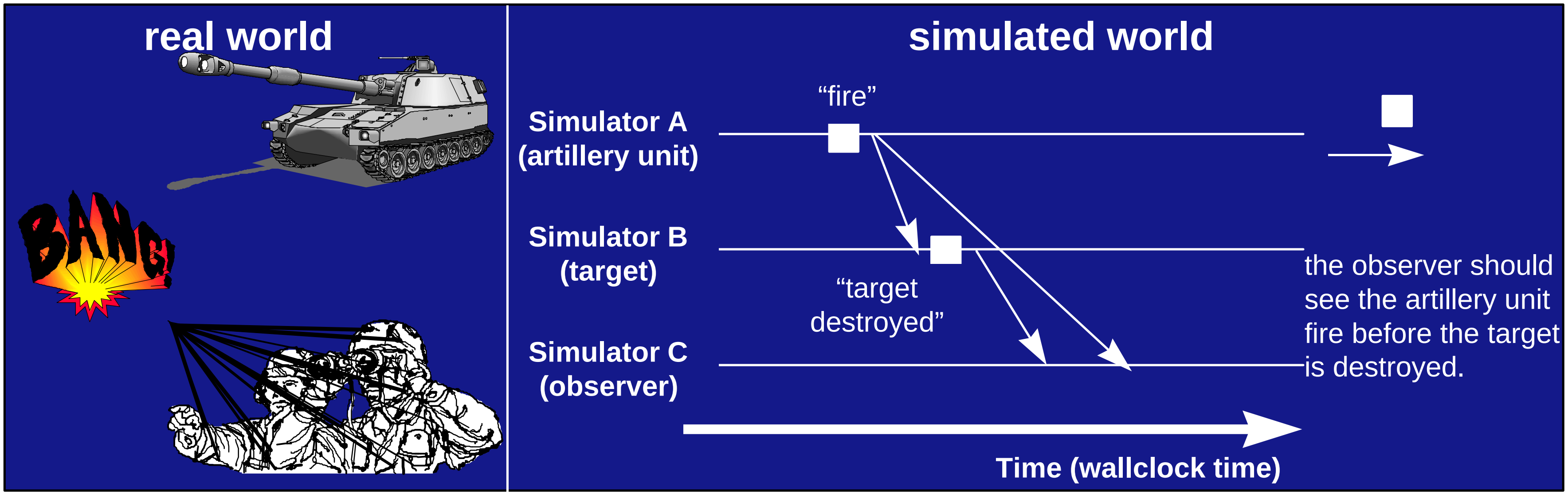
7.17 Is Attribute Owned By Federate

Time Management

- Control advancement of federates along with federation time
  - Coordinated with object management services to support Timestamp Ordered Delivery (causal behavior) across the federation if desired
  - Designed to support federates with different ordering and delivery requirements
  - Federates responsible for pacing to (scaled) real-time clock
- Interface functions include
  - Enable / Disable Time Regulated / Constrained
  - Query RTI for current values of time
    - Federate's logical time (LT), lower bound time stamp (LBTS), Minimum next event time
  - Modify and request lookahead values
  - Time Advance Request, Next Event Request and Flush Queue Request, and Time Advance Grant
  - Services for retracting events
  - Modification of default ordering type for instance attributes

Causality

- “Things” happen in the real world in a certain order (e.g., cause & effect)
- It should appear that events in the simulated world happen in the same order as the real world actions that they represent



real world

simulated world

Simulator A (artillery unit)

Simulator B (target)

Simulator C (observer)

“fire”

“target destroyed”

the observer should see the artillery unit fire before the target is destroyed.

Time (wallclock time)

- If the message for the “fire” event is delayed in the network, the observer will “see” the target is destroyed before the artillery unit fired upon it!
- Temporal anomalies such as this may or may not be acceptable, depending on the federation’s goals

Time Management Services

8.2 Enable Time Regulation

8.3 Time Regulation Enabled †

8.4 Disable Time Regulation

8.5 Enable Time Constrained

8.6 Time Constrained Enabled †

8.7 Disable Time Constrained

8.8 Time Advance Request

8.9 Time Advance Request Available

8.10 Next Event Request

8.11 Next Event Request Available

8.12 Flush Queue Request

8.13 Time Advance Grant †

8.14 Enable Asynchronous Delivery

8.15 Disable Asynchronous Delivery

8.16 Query LBTS

8.17 Query Federate Time

8.18 Query Minimum Next Event Time

8.19 Modify Lookahead

8.20 Query Lookahead

8.21 Retract

8.22 Request Retraction †

8.23 Change Attribute Order Type

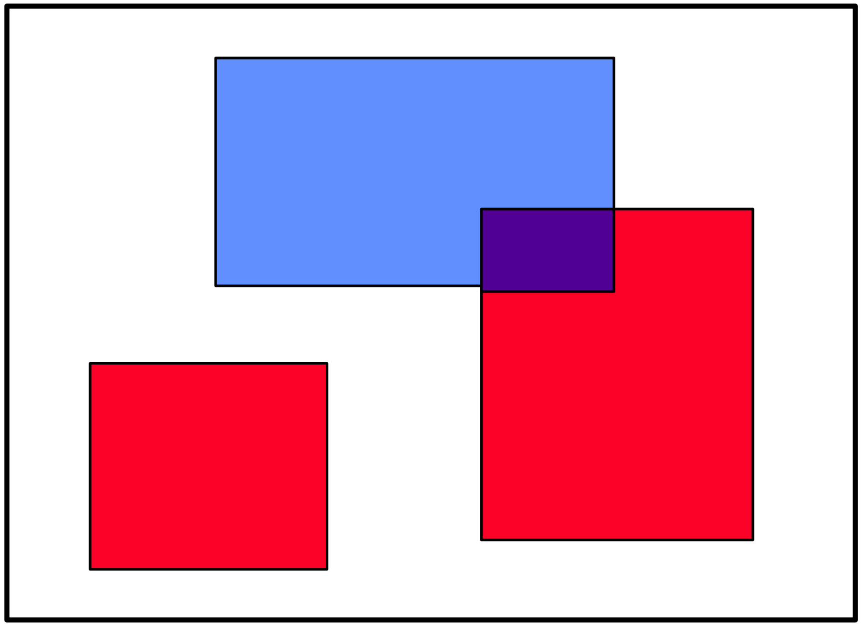
8.24 Change Interaction Order Type

Data Distribution Management

- Allow federates to specify the distribution conditions for the specific data they send or ask to receive
  - RTI uses this information to route data from producers to consumers based on DDM declarations
  - Not bound by FOM, data distribution can be managed based on other characteristics of objects important to particular federation execution
  - Federation design creates ‘routing spaces’ for use during runtime; these are specified at federation creation time in the Federation Execution Details (FED) file
- Interface functions include
  - Create, modify and delete regions to bound routing space
  - Associate/unassociate regions with specific object instances
  - Register Object instance attributes with Regions
  - Send interactions with regions
  - Change values of threshold of dimensions of a region

DDM Illustration

Two Dimensional Interest Space



🟦 Update Region

🟥 Subscription Region

🟪 Overlap Region - Published Data Sent to Subscribing Federate

Data Distribution Management Services

9.2 Create Region

9.3 Modify Region

9.4 Delete Region

9.5 Register Object Instance With Region

9.6 Associate Region For Updates

9.7 Unassociate Region For Updates

9.8 Subscribe Object Class Attributes With Region

9.9 Unsubscribe Object Class With Region

9.10 Subscribe Interaction Class With Region

9.11 Unsubscribe Interaction Class With Region

9.12 Send Interaction With Region

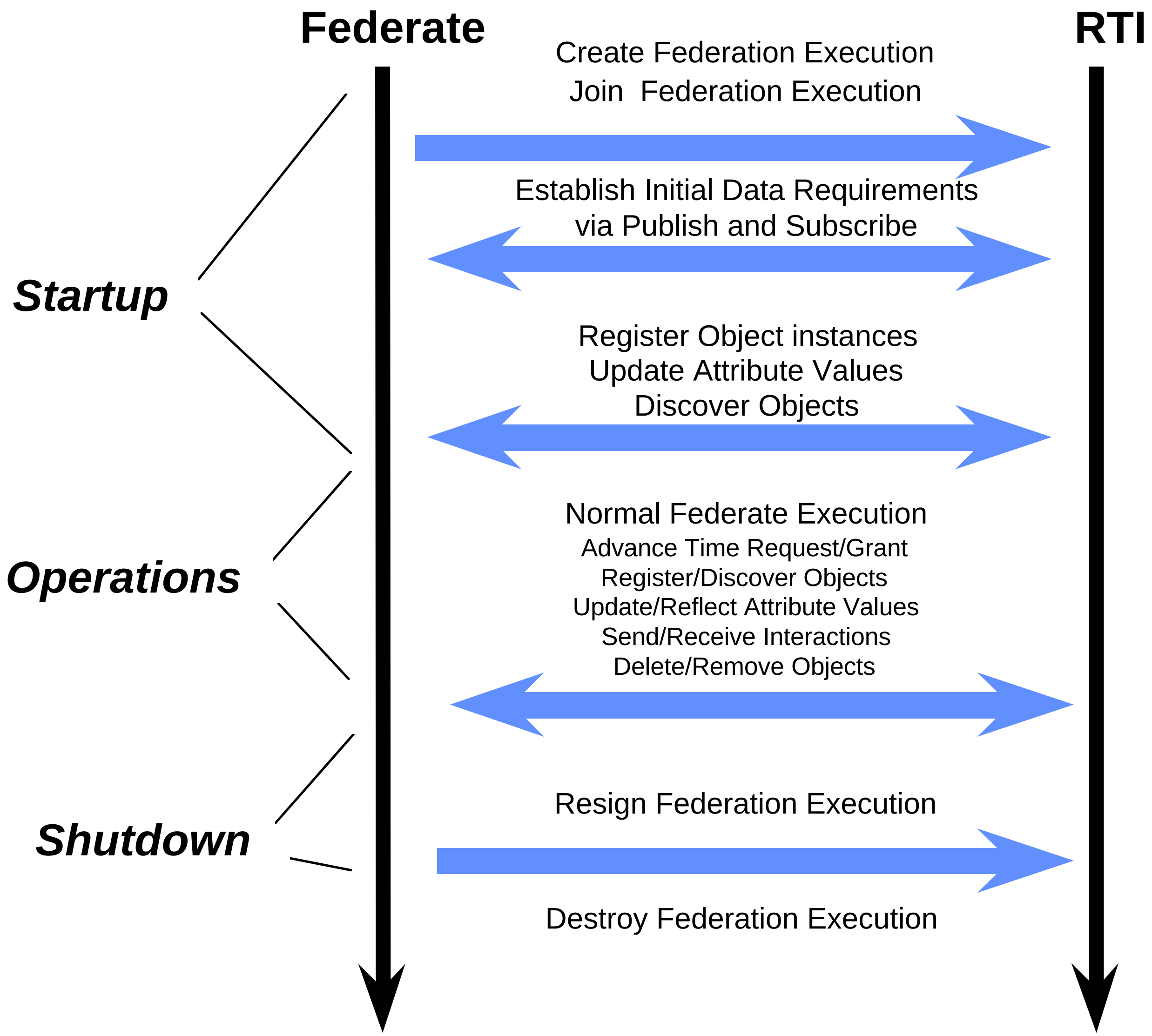
9.13 Request Attribute Value Update With Region

9.14 Change Thresholds †

Management Object Model

- Federation Executions are managed by a combination of Federate- and RTI-supplied information
- This information is structured using the same format (OMT) used for simulation data
- The MOM defines classes and interactions related to federation management just as the FOM defines classes and interactions in the simulation domain
- A manager federate can
  - Subscribe to MOM object classes and interactions exactly as it would to parts of any FOM
  - Monitor and control aspects of the federation through the MOM
- The RTI-supplied aspects of the MOM are being standardized
- Federate-supplied MOM data depends on the federation needs

Overview of Federation Execution Life Cycle


Federate
RTI
Create Federation Execution
Join Federation Execution
Establish Initial Data Requirements
via Publish and Subscribe
Startup
Register Object instances
Update Attribute Values
Discover Objects
Normal Federate Execution
Advance Time Request/Grant
Register/Discover Objects
Update/Reflect Attribute Values
Send/Receive Interactions
Delete/Remove Objects
Operations
Resign Federation Execution
Shutdown
Destroy Federation Execution

Summary

- **The Interface Specification along with the Rules and the Object Model Template comprise the technical specification of the High Level Architecture**
- **The Interface Specification provides a *STANDARD* specification of the functional interfaces between federates and any RTI**
  - **All RTIs are required to implement the full set of services specified in the document (and no more)**
  - **Particular Federates and Federations may or may not actually use the entire set of services**
  - **Services are provided as a toolbox from which individual federations select services to meet their particular needs**